MRP-Enabled PC-Based PROFINET Controller: Implementation and Performance Evaluation

İhsan Gökhun Önel
Siemens A.Ş
Istanbul, Türkiye
ihsan.onel@siemens.com

İlyas Karaca
Siemens A.Ş
Istanbul, Türkiye
ilyas.karaca@siemens.com

Ömer Korçak
Marmara University
Istanbul, Türkiye
omer.korcak@marmara.edu.tr

***Abstract*—This paper presents the integration of Media Redundancy Protocol (MRP) into PC-based PROFINET Controllers, addressing the lack of ring redundancy support on standard computing platforms. The proposed solution combines the PROFINET Driver development kit with the CP1625 PCIe communication card to enable MRP Manager functionality on a PC-based IO-Controller. We describe the system architecture, the role of the CP1625 embedded firmware in handling MRP independently from the host operating system, and the experimental evaluation of ring recovery performance. Packet captures taken at both ring ports during a link failure event show recovery times of approximately 44 ms and 30 ms, well within the 200 ms limit specified by IEC 62439-2. These results demonstrate that PC-based controllers can achieve network redundancy comparable to dedicated hardware controllers, offering system integrators the combined benefits of flexible PC-based control and reliable ring topology protection.**

***Index Terms*—Industrial Communications, PROFINET, Media Redundancy Protocol, MRP, PC-based Controller, Network Redundancy, IEC 62439**

I. INTRODUCTION

Ethernet-based networks have become a key communication backbone for industrial automation due to their performance and cost effectiveness. IEC 61784-2 defines profiles for industrial Real-Time Ethernet (RTE) networks and supports reliable and interoperable deployments of protocols such as PROFINET [1], [2]. In many installations, maintaining deterministic communication under failures is critical, and ring topologies are commonly adopted to increase availability.

The Media Redundancy Protocol (MRP), specified in IEC 62439-2, is the standard mechanism to provide ring redundancy in industrial Ethernet by enabling fast fault detection and reconfiguration [3]. In modern plants, where downtime directly translates into production losses, redundancy must be achieved without compromising real-time communication requirements. MRP defines two roles: an *MRP Manager*, which supervises ring status and coordinates recovery, and *MRP Clients*, which follow the manager's commands to update port states and maintain loop-free operation.

At the same time, industrial automation systems increasingly rely on PC-based PROFINET controllers for flexibility and computing capacity [4]. However, unlike dedicated hardware controllers, PC-based solutions typically lack native MRP Manager support. As a result, system integrators must either forgo ring redundancy or employ complex workarounds, creating an undesirable trade-off between PC-based control benefits and high availability.

This paper addresses this gap by enabling MRP support for a PC-based PROFINET IO-Controller using the PROFINET Driver development kit together with the CP1625 PCIe communication card [5], [6]. We describe the architecture and integration approach and evaluate ring recovery behavior through packet captures under controlled link-failure events, with respect to the IEC 62439-2 recovery requirement [3].

The main contributions of this paper are:

- An MRP-enabled PC-based PROFINET IO-Controller design that combines the PROFINET Driver development kit (PN Driver) with the CP1625 PCIe communication card to provide MRP Manager capability on standard PC platforms.
- A split architecture in which MRP supervision and recovery functions are executed on the CP1625 embedded PROFINET run-time environment, while the host performs PROFINET IO control via PN Driver.
- An experimental evaluation based on port-level packet captures under controlled link-failure events, reporting recovery behavior with respect to the IEC 62439-2 recovery requirement and the impact on cyclic RT communication.

The rest of the paper is organized as follows: Section II reviews related work, Section III describes the system architecture, Section IV details the MRP integration methodology, Section V presents the evaluation setup, and Section VI reports the results and discussion, followed by conclusions in Section VII.

II. RELATED WORK

Network redundancy is essential in industrial communication systems to sustain operation under link or device failures. In PROFINET networks, the Media Redundancy Protocol (MRP) defined in IEC 62439-2 is the standard mechanism for building redundant ring topologies [3]. The standard specifies that, after a single fault in the ring, communication should be restored within 200 ms for typical MRP operation [7].

Several studies have evaluated the performance of MRP in industrial environments. Giorgetti *et al.* [8] analyzed redundancy behavior in industrial Ethernet networks, providing

---

This work is supported by Siemens.

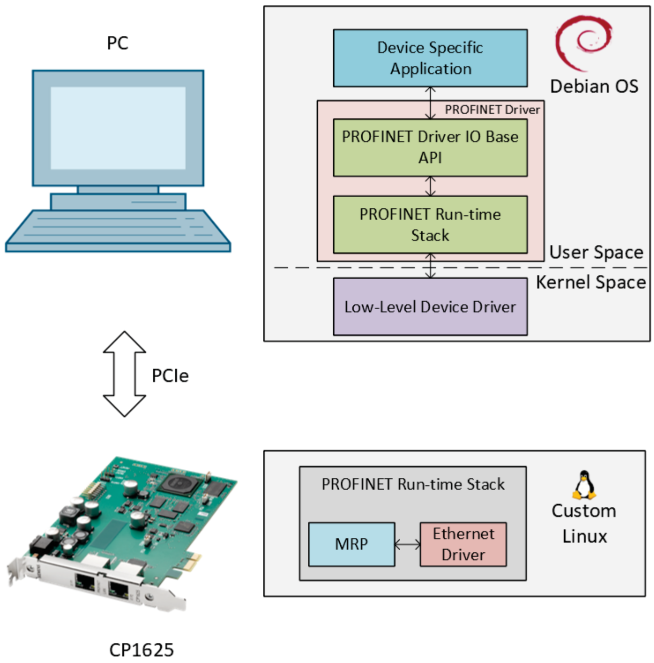


Fig. 1. System Architecture

useful insight into timing characteristics during fault and recovery events. A large-scale experimental study conducted in a real factory setting investigated a PROFINET ring network and reported a recovery time of 86 ms, which satisfies the IEC requirement [7]. That work also highlighted that jitter and timing variations can depend more on device characteristics than on physical factors such as cable length or the number of intermediate switches.

Felser [9] discussed redundancy options for PROFINET IO and outlined different implementation approaches for MRP, together with their potential impact on communication performance. In addition, Belić and Martinović [10] proposed a model to study the effect of MRP-related mechanisms on overall network performance, which is relevant for understanding additional traffic and timing implications. Prytz [11] provided comparative performance results across industrial Ethernet technologies (including PROFINET), offering broader context for expected real-time behavior.

While these studies provide valuable understanding of MRP in industrial networks, they mainly focus on setups using dedicated hardware controllers or embedded devices with built-in redundancy support. They do not specifically address how to realize MRP Manager capability in a PC-based PROFINET IO-Controller and evaluate its recovery behavior when redundancy functions are offloaded to a dedicated communication card. Our work targets this gap by enabling MRP support for a PC-based controller using the PROFINET Driver development kit together with the CP1625 card and by presenting a port-level packet-capture-based recovery evaluation.

III. SYSTEM ARCHITECTURE

The PROFINET Driver development kit (PN Driver) is a software package that enables developers to implement PROFINET communication functionality in PC-based applications [5]. It provides libraries and APIs that support integrating PROFINET Controller or Device functionality into standard computing platforms. With PN Driver, an application can communicate with PROFINET field devices, configure network parameters, and perform cyclic real-time data exchange. The kit includes documentation, sample code, and test utilities to support development and verification across different operating systems and hardware platforms. PN Driver supports standard Ethernet controllers as well as dedicated communication processors such as the CP1625 PCIe card, which provides hardware-accelerated PROFINET communication capabilities.

The CP1625 is a PCIe expansion board designed for SIMATIC industrial PCs and optimized for PROFINET IRT communication [6]. It features two RJ45 Ethernet ports that serve as the ring ports and supports 100 Mbit/s operation. The card integrates a real-time ASIC and a 2-port real-time switch with autonegotiation and autocrossing features, providing deterministic handling of PROFINET traffic. In addition, the CP1625 includes an embedded PROFINET run-time environment (firmware/stack) for time-critical communication functions close to the hardware.

Figure 1 illustrates how the host PC connects to the industrial network through the CP1625 card over PCIe. The host runs a Debian-based Linux operating system with user space and kernel space components. In user space, the device-specific application implements the control logic and interacts with the PN Driver IO Base API. Below this interface, the PN Driver run-time stack handles the core PROFINET protocol processing required for cyclic RT communication. In kernel space, the low-level device driver provides direct access to the CP1625 hardware and enables efficient data exchange between the host stack and the communication card.

On the CP1625 side, MRP functionality is executed within the card's embedded run-time environment. This allows ring supervision and reconfiguration to be handled close to the hardware, while the host application continues PROFINET IO operation and cyclic RT communication via PN Driver. This separation combines the flexibility of PC-based control with ring redundancy support provided through the CP1625 communication card.

IV. MRP INTEGRATION METHODOLOGY

This section summarizes the MRP operating principles relevant to our measurements and explains how MRP Manager functionality is realized by offloading ring supervision to the CP1625 communication card.

A. MRP Ring Operation in Closed and Open Topologies

MRP provides ring redundancy by ensuring loop-free operation under normal conditions while maintaining an alternative path for single-link failures [3]. Figure 2 shows a *closed ring* where all links are intact. When a physical link failure occurs,

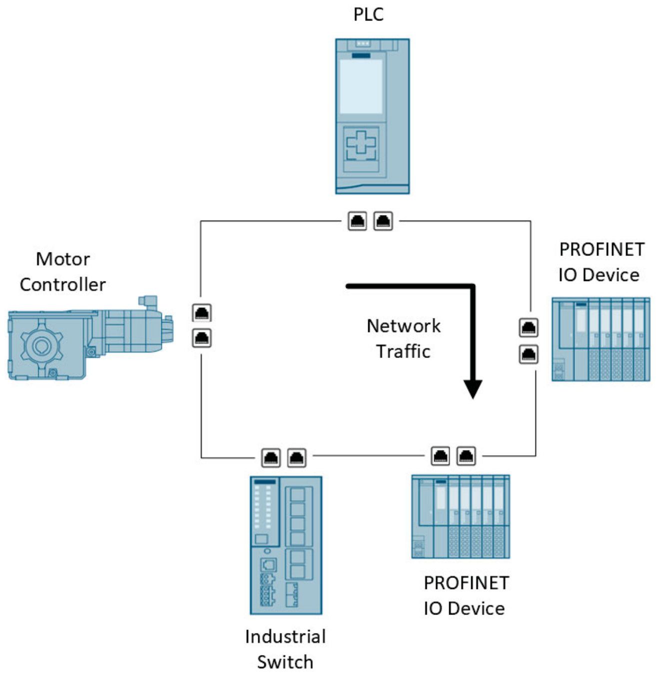


Fig. 2. MRP Ring Topology (closed ring).

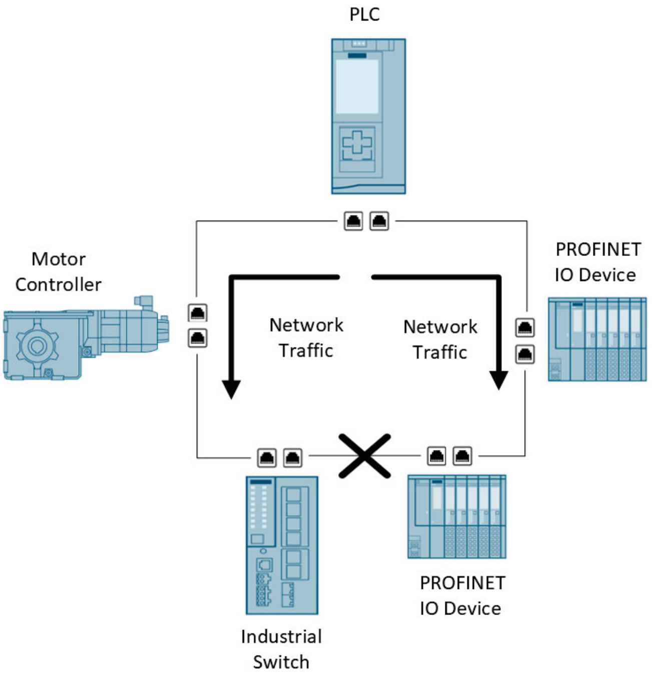


Fig. 3. MRP Topology with Broken Ring (open ring due to a link failure).

the ring becomes effectively *open* (Fig. 3). The MRP Manager detects the disruption and triggers a topology change such that forwarding/blocking behavior is updated and cyclic RT traffic is rerouted through the alternative direction, restoring loop-free connectivity.

B. MRP Operation and Detection Parameters

In MRP, a single MRP Manager supervises the ring by periodically transmitting MRP test frames on both ring ports and observing their return behavior through participating MRP Clients [3]. The test frame interval is governed by TST_{default}. If expected test frames do not return before the internal timer expires, a missed-return counter is incremented. When the number of consecutive missed returns reaches $TSTNR_{\max}$, the manager declares a ring failure and initiates topology reconfiguration. Accordingly, the failure detection time can be expressed as [7]:

$$T_{\text{detection}} = TST_{\text{default}} \times TSTNR_{\max}. \tag{1}$$

In this implementation, $TSTNR_{\max}$ is set to 10, the default value defined by IEC 62439-2 [3]. With a TST_{default} of 20 ms, this gives a maximum detection time of $20 \times 10 = 200$ ms, which is the standard limit for MRP class 1. This value allows fast fault detection while avoiding false alarms from brief frame delays.

C. MRP Integration with PN Driver and CP1625

The host PC provides IO-Controller functionality via PN Driver, including configuration and cyclic RT data exchange with IO-Devices [5]. MRP Manager functionality is executed on the CP1625 embedded firmware/stack. Upon a link failure, the CP1625 detects missing test frame returns and performs topology change processing by emitting MRP control frames and updating ring port forwarding/blocking behavior. During this transient, the host application continues its IO tasks via PN Driver while the CP1625 restores a loop-free forwarding state through the alternative ring direction.

D. Measurement Definition and Capture Method

Packet captures were recorded at both CP1625 ring ports during controlled physical link-failure events. We operationally define the ring recovery time T_{recovery} as the elapsed time between (i) the last RT cyclic frame observed immediately before the failure-induced disruption and (ii) the first RT cyclic frame observed once the rerouted path becomes active at the monitored port. This definition captures the effective communication gap visible at each port.

Cyclic communication properties were further characterized using RT frame inter-arrival times to estimate average cycle time and jitter before and after recovery. The resulting T_{recovery} values are compared against the IEC 62439-2 requirement (200 ms) and discussed together with observed effects on cyclic communication in the subsequent sections.

V. PERFORMANCE EVALUATION

A. Test environment setup

The experimental setup in Fig. 4 evaluates MRP behavior of the PC-based PROFINET IO-Controller in a ring topology. A SIMATIC Industrial PC equipped with a CP1625 PCIe card serves as the IO-Controller and operates as the MRP Manager.

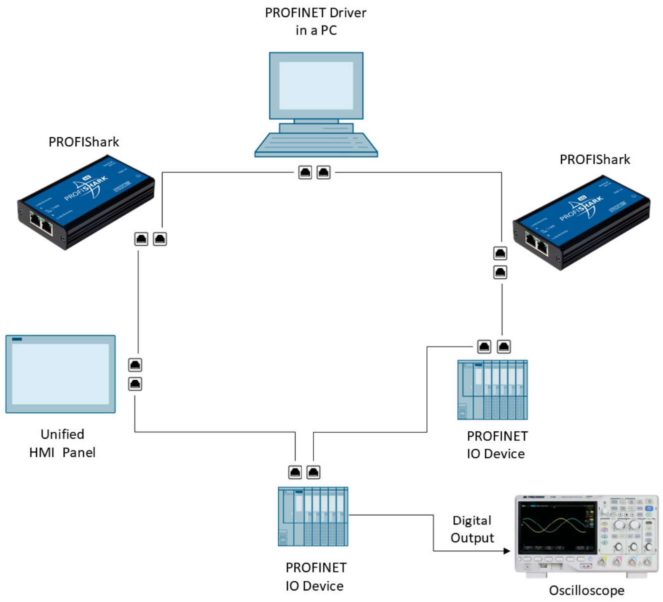


Fig. 4. Test Setup

The two RJ45 ports of the CP1625 act as ring ports and connect to a set of PROFINET IO-Devices acting as MRP Clients. The IO-Devices are connected in a daisy-chain to form a closed ring, with the first and last devices connected back to the two CP1625 ports. Devices were configured using Totally Integrated Automation (TIA) Portal [12] engineering software.

The host PC runs a Debian-based Linux operating system with PROFINET Driver development kit installed [5]. Link failures were introduced by physically disconnecting an Ethernet segment during operation. For each failure event, packet captures were collected at both ring ports to compute T_{recovery} (as defined in Section IV) and to verify the continuity and timing behavior of cyclic RT communication after ring reconfiguration.

B. Evaluation metrics

To assess the MRP integration on the PC-based IO-Controller, we use the following metrics, selected to align with IEC 62439-2 [3] and prior work [7], [8], [10]:

- **Ring Recovery Time** (T_{recovery}): The elapsed time between a link failure and the re-establishment of cyclic communication through the alternative path. According to IEC 62439-2, this value must not exceed 200 ms for standard MRP operation.
- **Number of Lost Cyclic Frames** (N_{lost}): The number of PROFINET IO cyclic frames not delivered during recovery, estimated as:

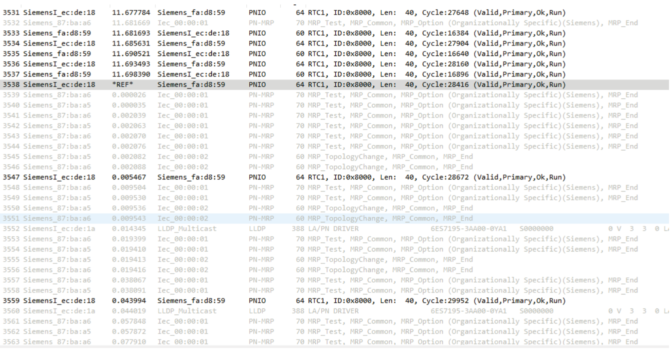
$$N_{\text{lost}} = \frac{T_{\text{recovery}}}{T_{\text{cycle}}} \tag{2}$$

  where T_{cycle} is the configured IO update cycle time.
- **Cyclic Data Jitter** (J): Variation in cyclic timing, measured as the standard deviation of RT inter-arrival times during steady-state and around recovery transients.
- **MRP Protocol Overhead**: Additional bandwidth consumption due to MRP test/control frames relative to cyclic traffic, influenced by TST_{default} and ring size.

All measurements were extracted from packet timestamps captured at both CP1625 ring ports with millisecond-level precision.

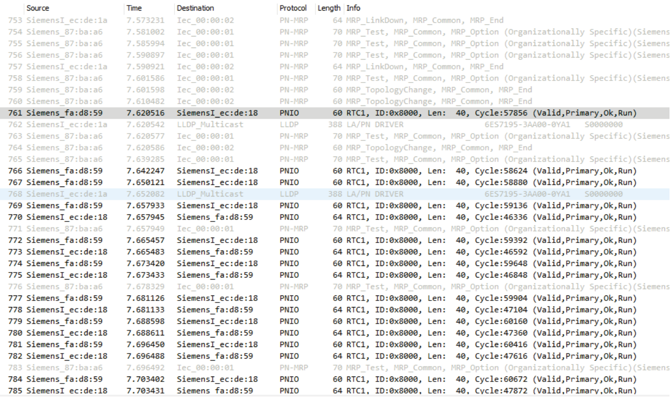
VI. RESULTS

Packet captures collected at the CP1625 ring ports provide empirical evidence that the proposed PC-based PROFINET IO-Controller with MRP support satisfies the recovery requirement of IEC 62439-2 [3]. Figure 5 and Fig. 6 show the Wireshark traces for two representative link-failure experiments captured at Port 1 and Port 2, respectively.



```
3531 SiemensI_ec:de:18   11.677784 Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:27648 (Valid,Primary,Ok,Run)
3532 Siemens_87:ba:a6    11.681669 Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3533 Siemens_fa:d8:59    11.681693 SiemensI_ec:de:18   PNIO      60 RTC1, ID:0x8000, Len:  40, Cycle:16384 (Valid,Primary,Ok,Run)
3534 SiemensI_ec:de:18   11.685631 Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:27904 (Valid,Primary,Ok,Run)
3535 Siemens_fa:d8:59    11.690521 SiemensI_ec:de:18   PNIO      60 RTC1, ID:0x8000, Len:  40, Cycle:16640 (Valid,Primary,Ok,Run)
3536 SiemensI_ec:de:18   11.693493 Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:28160 (Valid,Primary,Ok,Run)
3537 Siemens_fa:d8:59    11.698390 SiemensI_ec:de:18   PNIO      60 RTC1, ID:0x8000, Len:  40, Cycle:16896 (Valid,Primary,Ok,Run)
3538 SiemensI_ec:de:18   *REF*     Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:28416 (Valid,Primary,Ok,Run)
3539 Siemens_87:ba:a6    0.000026  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3540 Siemens_87:ba:a5    0.000035  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3541 Siemens_87:ba:a6    0.002039  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3542 Siemens_87:ba:a5    0.002063  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3543 Siemens_87:ba:a6    0.002070  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3544 Siemens_87:ba:a5    0.002076  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3545 Siemens_87:ba:a5    0.002082  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3546 Siemens_87:ba:a6    0.002088  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3547 SiemensI_ec:de:18   0.005467  Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:28672 (Valid,Primary,Ok,Run)
3548 Siemens_87:ba:a6    0.009504  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3549 Siemens_87:ba:a5    0.009530  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3550 Siemens_87:ba:a5    0.009536  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3551 Siemens_87:ba:a6    0.009543  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3552 SiemensI_ec:de:1a   0.014345  LLDP_Multicast      LLDP     388 LA/PN DRIVER                 6ES7195-3AA00-0YA1   S0000000          0 V  3  3  0 L
3553 Siemens_87:ba:a6    0.019399  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3554 Siemens_87:ba:a5    0.019410  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3555 Siemens_87:ba:a5    0.019413  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3556 Siemens_87:ba:a6    0.019416  Iec_00:00:02        PN-MRP    60 MRP_TopologyChange, MRP_Common, MRP_End
3557 Siemens_87:ba:a6    0.038067  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3558 Siemens_87:ba:a5    0.038091  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3559 SiemensI_ec:de:18   0.043994  Siemens_fa:d8:59    PNIO      64 RTC1, ID:0x8000, Len:  40, Cycle:29952 (Valid,Primary,Ok,Run)
3560 SiemensI_ec:de:1a   0.044019  LLDP_Multicast      LLDP     388 LA/PN DRIVER                 6ES7195-3AA00-0YA1   S0000000          0 V  3  3  0 L
3561 Siemens_87:ba:a6    0.057848  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3562 Siemens_87:ba:a5    0.057872  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
3563 Siemens_87:ba:a6    0.077910  Iec_00:00:01        PN-MRP    70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens), MRP_End
```

Fig. 5. Wireshark capture of MRP ring failure and recovery at Port 1 (Frames 3538–3560+)



```
    Source             Time       Destination         Protocol  Length Info
753 SiemensI_ec:de:1a  7.573231   Iec_00:00:02        PN-MRP        64 MRP_LinkDown, MRP_Common, MRP_End
754 Siemens_87:ba:a6   7.581002   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
755 Siemens_87:ba:a6   7.585994   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
756 Siemens_87:ba:a6   7.590897   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
757 SiemensI_ec:de:1a  7.590921   Iec_00:00:02        PN-MRP        64 MRP_LinkDown, MRP_Common, MRP_End
758 Siemens_87:ba:a6   7.601586   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
759 Siemens_87:ba:a6   7.601598   Iec_00:00:02        PN-MRP        60 MRP_TopologyChange, MRP_Common, MRP_End
760 Siemens_87:ba:a6   7.610482   Iec_00:00:02        PN-MRP        60 MRP_TopologyChange, MRP_Common, MRP_End
761 Siemens_fa:d8:59   7.620516   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:57856 (Valid,Primary,Ok,Run)
762 SiemensI_ec:de:1a  7.620542   LLDP_Multicast      LLDP         388 LA/PN DRIVER                 6ES7195-3AA00-0YA1   S0000000
763 Siemens_87:ba:a6   7.620577   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
764 Siemens_87:ba:a6   7.620586   Iec_00:00:02        PN-MRP        60 MRP_TopologyChange, MRP_Common, MRP_End
765 Siemens_87:ba:a6   7.639285   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
766 Siemens_fa:d8:59   7.642247   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:58624 (Valid,Primary,Ok,Run)
767 Siemens_fa:d8:59   7.650121   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:58880 (Valid,Primary,Ok,Run)
768 SiemensI_ec:de:1a  7.652082   LLDP_Multicast      LLDP         388 LA/PN DRIVER                 6ES7195-3AA00-0YA1   S0000000
769 Siemens_fa:d8:59   7.657933   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:59136 (Valid,Primary,Ok,Run)
770 SiemensI_ec:de:18  7.657945   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:46336 (Valid,Primary,Ok,Run)
771 Siemens_87:ba:a6   7.657949   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
772 Siemens_fa:d8:59   7.665457   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:59392 (Valid,Primary,Ok,Run)
773 SiemensI_ec:de:18  7.665483   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:46592 (Valid,Primary,Ok,Run)
774 Siemens_fa:d8:59   7.673420   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:59648 (Valid,Primary,Ok,Run)
775 SiemensI_ec:de:18  7.673433   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:46848 (Valid,Primary,Ok,Run)
776 Siemens_87:ba:a6   7.678329   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
777 Siemens_fa:d8:59   7.681126   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:59904 (Valid,Primary,Ok,Run)
778 SiemensI_ec:de:18  7.681133   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:47104 (Valid,Primary,Ok,Run)
779 Siemens_fa:d8:59   7.688598   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:60160 (Valid,Primary,Ok,Run)
780 SiemensI_ec:de:18  7.688611   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:47360 (Valid,Primary,Ok,Run)
781 Siemens_fa:d8:59   7.696450   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:60416 (Valid,Primary,Ok,Run)
782 SiemensI_ec:de:18  7.696488   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:47616 (Valid,Primary,Ok,Run)
783 Siemens_87:ba:a6   7.696492   Iec_00:00:01        PN-MRP        70 MRP_Test, MRP_Common, MRP_Option (Organizationally Specific)(Siemens
784 Siemens_fa:d8:59   7.703402   SiemensI_ec:de:18   PNIO          60 RTC1, ID:0x8000, Len:  40, Cycle:60672 (Valid,Primary,Ok,Run)
785 SiemensI_ec:de:18  7.703431   Siemens_fa:d8:59    PNIO          64 RTC1, ID:0x8000, Len:  40, Cycle:47872 (Valid,Primary,Ok,Run)
```

Fig. 6. Wireshark capture of MRP ring failure and recovery at Port 2 (Frames 757–770+)

A. Ring Recovery Behavior

During normal operation, PROFINET RT cyclic frames (EtherType 0x8892) are exchanged between the IO-Controller and IO-Devices with an average cycle time of about 8.1 ms at Port 1. In parallel, the MRP Manager sends periodic test frames (EtherType 0x88E3) every 20 ms from both ring ports.

In the Port 1 trace (Fig. 5), the following sequence of *observable* events is seen after a physical link disconnection:

1) **Last RT frame before the transient (Frame 3538)**: The last RT cyclic response observed at the Port 1 capture point before the failure-induced transient.
2) **MRP control activity appears (Frames 3539–3546)**: Within a few milliseconds (≈2.1 ms) after the last observed RT frame at the capture point, a burst of MRP control frames is observed on the ring, including topology-change-related messages. This indicates that the MRP Manager is entering the reconfiguration phase following the disruption.
3) **Reconfiguration transient (Frames 3547–3558)**: MRP control/test traffic continues while the ring transitions to the new path and reaches a loop-free forwarding state. A small number of RT frames may still be observed at Port 1 during this transient due to in-flight traffic and convergence effects.
4) **Traffic ceases at Port 1 (Frame 3559)**: After $\Delta t \approx$ 43.99 ms (measured from Frame 3538 to Frame 3559), RT cyclic frames are no longer observed at the Port 1 capture point, indicating that cyclic traffic has been rerouted away from Port 1 and is now carried via the alternative ring direction.
5) **Post-recovery steady state (Frames 3560+)**: After convergence, only periodic MRP test frames (every ∼20 ms) are observed at the original capture point, consistent with cyclic data being forwarded through the alternative path.

A separate capture was taken at Port 2 of the CP1625 card during a different link failure test (Fig. 6). Before the failure, only MRP test frames (every ∼20 ms) are visible at Port 2, consistent with steady-state cyclic traffic flowing via Port 1. After the ring reconfiguration, RT cyclic frames begin to appear at Port 2; the gap between the last RT frame before the transient (Frame 757) and the first RT frame after the rerouted path becomes active (Frame 761) is $\Delta t \approx 29.60$ ms. This observation confirms that cyclic traffic is successfully redirected through the alternative ring direction after a failure.

B. Compliance with IEC 62439-2

The recovery gaps measured from the two representative tests are approximately 44 ms at Port 1 and 30 ms at Port 2, both well below the 200 ms limit specified by IEC 62439-2 [3]. These values are also lower than the 86 ms recovery time reported in a real factory test in [7]. While recovery time depends on the ring size and the configured TST_{default} and $TSTNR_{\text{max}}$ values, our measurements provide a margin of more than 150 ms below the standard limit under the tested configuration. The implementation is therefore expected to remain compliant for larger rings or higher traffic loads, although recovery time may increase with scale and load.

C. Cyclic Communication Impact

The IO cycle time averaged 8.1 ms at Port 1 during normal operation and 7.8 ms at Port 2 after recovery. The measured jitter values were 0.56 ms (Port 1) and 0.76 ms (Port 2). The largest inter-arrival gap between two consecutive RT frames was 8.87 ms at Port 1 and 21.73 ms at Port 2 (the first cycle immediately after switchover). Overall, the captures indicate that the redundancy action does not introduce persistent disruption to cyclic RT communication. During the ≈44 ms transient observed at Port 1, an estimated $N_{\text{lost}} \approx \lceil 43.99/8.1 \rceil = 6$ cyclic frames may have been missed. Such a short gap is often tolerable in many deployments, since PROFINET IO watchdog/timeout settings typically allow a configurable number of missed cyclic updates before raising a diagnostic event.

D. MRP Control Traffic During Recovery

During steady state, the MRP Manager emits test frames periodically in pairs (one from each ring port) every 20 ms. Around the failure event, a short burst of additional MRP control frames is observed at the capture point (about 8 frames within ≈2.1 ms after the last observed RT frame), including topology-change-related messages, until the ring converges to the new forwarding state. This increase in MRP control traffic is short-lived and no persistent impact is observed on cyclic RT communication after convergence.

E. Results Summary

Table I summarizes the key performance metrics. Note that the Port 1 and Port 2 recovery values are obtained from separate link-failure tests captured at different ports.

TABLE I. SUMMARY OF MEASURED PERFORMANCE METRICS.

Metric	**Port 1**	**Port 2**	**IEC Limit**
Recovery gap	44 ms	30 ms	200 ms
Avg. cycle time	8.1 ms	7.8 ms	—
Jitter (σ)	0.56 ms	0.76 ms	—
Max. frame gap	8.87 ms	21.73 ms	—
Est. lost frames	6	—	—
MRP test interval	20 ms		—
Compliance	**Pass**	**Pass**	—

VII. DISCUSSION

A. Comparison with Prior Work

Table II compares the recovery performance of our PC-based implementation with results reported in previous studies.

TABLE II. COMPARISON WITH PRIOR WORK

Study	**Recovery Time**	**Platform**
IEC 62439-2 limit [3]	200 ms	Standard
ProfinetRingStudy [7]	86 ms	HW controller
Giorgetti et al. [8]	Not reported	HW controller
This work (Port 1)	**44 ms**	PC-based
This work (Port 2)	**30 ms**	PC-based

The recovery gaps observed in our experiments are below the 86 ms reported by [7] in a factory environment with a larger ring. This indicates that offloading MRP functions to the CP1625 embedded firmware can achieve recovery performance comparable to dedicated hardware controllers under the tested configuration. Differences across studies may be attributed to ring size, the number of MRP Clients, background traffic, and the configured TST_{default} and $TSTNR_{\text{max}}$ parameters; a direct comparison under identical conditions would be needed for definitive conclusions.

B. Advantages of the PC-Based Approach

The results indicate that running the MRP logic on the CP1625 firmware is an effective design choice. MRP detection and topology-change processing are handled on the CP1625's embedded PROFINET run-time stack without requiring host application intervention in the reconfiguration path. This keeps the host CPU available for the main control application, and makes recovery behavior primarily dependent on the CP1625 firmware rather than host-side Linux scheduling, which may introduce variability.

C. Limitations

This evaluation has several limitations. First, the presented results are based on representative link failure events; more extensive validation would require repeated trials under different ring sizes and traffic conditions. Second, although captures were taken at both ring ports, additional capture points within the ring (e.g., at IO-Devices) would provide a more complete view of recovery impact across devices. Third, the test setup used a limited number of IO-Devices; testing with larger rings containing more MRP Clients is needed to assess scalability.

VIII. CONCLUSION AND FUTURE WORK

This paper presented the integration of MRP support for a PC-based PROFINET IO-Controller using the PROFINET Driver development kit and the CP1625 PCIe communication card. The proposed design executes MRP functions on the CP1625 embedded firmware/stack, reducing dependence on host-side scheduling effects. Packet-capture-based experiments at the ring ports show recovery gaps of approximately 44 ms (Port 1) and 30 ms (Port 2), which are well below the 200 ms limit specified by IEC 62439-2. These results demonstrate that PC-based controllers can achieve standards-compliant ring redundancy, combining the flexibility of PC-based control with robust availability in ring topologies.

Future work includes scalability testing with larger rings and different traffic conditions, statistical analysis over repeated failure–recovery trials, investigating MRP Manager support on standard Ethernet adapters without CP1625, and extending the implementation to support seamless redundancy concepts such as MRPD (Media Redundancy for Planned Duplication) and multi-ring topologies.

ACKNOWLEDGMENT

We extend our gratitude to Siemens Sanayi ve Tic. A.S. for their invaluable support.

REFERENCES

[1] *Industrial communication networks - Profiles*, International Electrotechnical Commission Std. IEC 61 784, 2019. [Online]. Available: https://webstore.iec.ch/publication/6029

[2] *Industrial communication networks - Fieldbus specifications*, International Electrotechnical Commission Std. IEC 61 158, 2019. [Online]. Available: https://webstore.iec.ch/publication/6028

[3] *IEC 62439-2:2016 Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP)*, International Electrotechnical Commission Standard, 2016.

[4] M. Felser, "Real-time ethernet - industry prospective," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118–1129, 2005.

[5] S. AG. (2023) Profinet driver development kit for profinet controller and device. Siemens Industry Online Support. [Online]. Available: https://support.industry.siemens.com/cs/document/109817202/profinet-driver-development-kit-for-profinet-controller-and-device?dti=0&lc=en-FI

[6] ——, *CP 1625 / CP 1625Dev Operating Instructions*, Siemens AG, Germany, January 2023, document Number: A5E43075902-AC.

[7] G. S. Sestito, A. C. Turcato, A. L. Dias, M. S. Rocha, D. Brandão, and R. da Veiga Torres, "Case of study of a profinet network using ring topology," in *2016 IEEE International Symposium on Consumer Electronics (ISCE)*. IEEE, 2016, pp. 91–96.

[8] A. Giorgetti, F. Cugini, F. Paolucci, L. Valcarenghi, M. Pioro, and P. Castoldi, "Performance analysis of media redundancy protocol (mrp)," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 218–227, 2013.

[9] M. Felser, "Media redundancy for PROFINET IO," in *7th IEEE International Workshop on Factory Communication Systems*, Dresden, Germany, May 2008, pp. 325–330.

[10] F. Belić and G. Martinović, "Model of influence of MRP on network performance," in *18th IEEE Symposium on Computers and Communications (ISCC)*, Split, Croatia, July 2013, pp. 100–105.

[11] G. Prytz, "A performance analysis of EtherCAT and PROFINET IRT," in *IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 408–415.

[12] Siemens. (2024, Jun) Totally integrated automation portal. Accessed: Jun. 1, 2024. [Online]. Available: https://www.siemens.com/global/en/products/automation/industry-software/automation-software/tia-portal.html